A mostly-traditional approach improves alignment of bisulfite-converted DNA: Supplement

Martin C. Frith, Ryota Mori and Kiyoshi Asai

1 Supplementary methods

1.1 Using sequence quality data

Suppose we wish to align a sequence with quality data (e.g. a DNA read) to a sequence without quality data (e.g. a genome). We previously showed that the log-likelihood-ratio score for aligning base x in the genome with position d in the read is as follows [2]:

$$S'_{xd} = T \ln \left(\sum_y \frac{M_{xy}}{A_x B_y} P(y|d) \right)$$

Here, $P(y|d)$ is the probability that the base at position d is really y. Sometimes, the sequencer data directly tells us $P(y|d)$ (e.g. PRB format). Usually, however, we are given just one base (z) and one error probability (e) per position (e.g. FASTQ format).

1.1.1 Old method

Previously, we inferred $P(y|d)$ like this:

$$P(y|d) = \begin{cases} 1 - e & \text{if } y = z, \\ e/3 & \text{if } y \neq z. \end{cases}$$

In other words, we split the error equally among the other 3 bases, which seems reasonable when they are equally abundant.

1.1.2 New method

We have modified this so as to split the error among the other 3 bases in proportion to their abundances:

$$P(y|d) = \begin{cases} 1 - e & \text{if } y = z, \\ eB_y/(1 - B_z) & \text{if } y \neq z. \end{cases}$$

The new method requires us to know the base abundances B_y. In our implementation, we do not infer them from the sequence data, but rather we infer them from the score matrix. This has the advantage that the scoring scheme does not fluctuate confusingly whenever the sequence data changes.

1.1.3 Formula simplification

The new method leads to a simple and intuitive formula for S'_{xd}. First, define quantities that we call "uncertainty" (u) and "certainty" (c):

$$u = \frac{e}{1 - B_z} \qquad\qquad c = 1 - \frac{e}{1 - B_z}$$

The intuition behind these definitions is that "error" means that the base is definitely one of the other three, whereas "uncertainty" means that the base could be any of the four in proportion to their abundances. In any case, the formula for S'_{xd} simplifies to the following:

$$S'_{xd} = T \ln \left(c \frac{M_{xy}}{A_x B_y} + (1 - c) \right)$$

1.1.4 When both sequences have quality data

If we align two sequences that both have quality data, the log-likelihood-ratio score is as follows [3]:

$$S''_{d_i d_2} = T \ln \left(\sum_{x,y} \frac{M_{xy}}{A_x B_y} P(x|d_1) P(y|d_2) \right)$$

This formula also simplifies greatly:

$$S''_{d_1 d_2} = T \ln \left(c_1 c_2 \frac{M_{xy}}{A_x B_y} + (1 - c_1 c_2) \right)$$

1.2 Settings for the tested alignment methods

1.2.1 Bowtie/Bismark

Bismark v0.5.3 was executed as follows:

```
bismark_genome_preparation --verbose workdir
cd workdir
trimEnd -q $q reads.fastq > trimmed.fastq
bismark --directional -n $n -l $l workdir trimmed.fastq
```

We tried several values for q, n, and l (Supplementary dataset 2). Figures 2,3,5 show one combination that performed well: q=3, n=2, l=50.

The trimEnd script (not part of Bismark) trims the 3' end of each read to just before the first base with phred score $< q$.

1.2.2 Bowtie/BS_Seeker

BS_Seeker was executed as follows:

```
python Preprocessing_genome.py -f genome.fa -t N -p bowtie-0.12.7/ >
 log_Preprocessing_genome.txt
python BS_Seeker.py -i reads.fastq -t N -e $e -p bowtie-0.12.7/ -m $m -o out.txt
```

We tried several values for e and m (Supplementary dataset 2). Figures 2,3,5 show one combination that performed well: e=50, m=3.

1.2.3 Bowtie/Lister

Bowtie 0.12.7 was used following the recipe in Lister et al. 2009 [4]. This includes application of trimEnd with q=3. The adapter-trimming step, however, was omitted, since our benchmark lacks adapters.

1.2.4 Brat

Brat 1.2.2 was executed as follows:

```
trim -s reads.fastq -P myPrefix -q $q -m 2
brat -r fastaFileNames.txt -s myPrefix_reads1.txt -bs -m $m -f $f -o output.txt
```

We tried several values for q, m, and f (Supplementary dataset 2).

1.2.5 Bsmap

Bsmap 2.2 was executed as follows:

```
bsmap -a reads.fastq -d genome.fa -o outFile -v $v -s $s -w $w
```

We tried several values for v, s, and w (Supplementary dataset 2).

We tried two ways of dealing with non-unique maps. We either used w=2 and discarded output with map_flag $\neq$ UM, or used w=100 and kept output with map_flag = UM or MA or OF.

Figures 2,3,5 show one combination that performed well: v=10, s=16, w=2.

1.2.6 Gsnap/MethylCoder

MethylCoder 0.3.8 was executed as follows:

```
methylcoder --gsnap gmap-2011-03-28.v3/bin --outdir myOutDir
 --extra-args ‘‘--quiet-if-excessive --npaths 1’’ --mismatches=2
 --reference genome.fa reads.fastq
```

1.2.7 Novoalign

Novoalign V2.07.17 was executed as follows:

```
novoindex -b genome.nbx genome.fa
novoalign -b2 -c1 -t $t --Q2Off -d genome.nbx -f reads.fastq
```

We tried several values for t (Supplementary dataset 2). We also tried omitting the t option.

1.2.8 Pash

Pash 3.0.6.2 was executed as follows:

```
getRCChrom.rb genome.fa meth.fa
keyFreq.exe -o kmerCounts -p $p meth.fa
makeIgnoreList.exe -i kmerCounts -o meth.il -c $c
pash-3.0lx.exe -h meth.fa -v reads.fastq -G $g -k $k -n $n -o out.pash
 -s 30 -d $d -S /tmp -B -L meth.il
```

We tried a few different values for d, g, k, n, p, and c (Supplementary dataset 2).

We tried two seeding strategies. We used either: p=111011011000110101011, k=13, n=21, c=367; or: p=111010110100110111, k=12, n=18, c=1456. In each case, the value of c was chosen so as to discard 5% of kmers.

Figures 2,3,5 show: g=1, d=800, k=13, n=21, p=111011011000110101011, c=367.

1.2.9 Rmap

Rmap v2.05 was executed as follows:

```
rmapbs -v -Q -B -F chromosomes_file_set.txt reads.fastq
```

1.2.10 Last

Last version 192 was executed as follows:

```
lastdb -u bisulfite_f.seed fIndex genome.fa
lastdb -u bisulfite_r.seed rIndex genome.fa
lastal -p bisulfite_f.mat -s1 -Q1 -j1 -d120 -f0 fIndex reads.fastq > fOut
lastal -p bisulfite_r.mat -s0 -Q1 -j1 -d120 -f0 rIndex reads.fastq > rOut
last-merge-batches.py fOut rOut | last-map-probs.py -s150 -m0.1
```

For gapped alignment, we replaced `-j1 -d120` with `-d108 -e120`. We varied the max seed frequency by using `lastal`'s `-m` option.

1.3 Measuring CPU time

Each CPU time is the sum of "user" and "sys" from the `time` command. All tests were performed on 2.53GHz Intel Xeon E5540 CPUs.

For methods such as Bismark that wrap an aligner (e.g. Bowtie), we wished to measure the time used by the aligner only. We did so directing the wrapper program to a fake aligner, which runs and times the real aligner.

1.4 Incorporating sequence context into alignment scores

Methylation rates typically depend on sequence context. For example, in mammal genomes, cytosine methylation occurs at CpGs more often than not. Plants have not only enzymes that methylate cytosines in CpG context, but those that methylate cytosines in CpHpG context. Here, adopting different F values F_1, F_2, and F_3 for Cs in CpGpN, CpHpG, and CpHpH context respectively, we may improve the alignment accuracy.

There are basically two strategies, according to which sequence context we focus on: query or reference. If we focus on query context, our aim is estimating F_{est} at each query position, as follows:

$$\begin{aligned} F_{est} &= \mathbb{E}\left[F|\mathrm{N}_1,\mathrm{N}_2\right] \\ &= \sum_{i=1}^{3} F_i P(F_i|\mathrm{N}_1,\mathrm{N}_2) \end{aligned}$$

$\mathbb{E}[F|\mathrm{N}_1, \mathrm{N}_2]$ is the expectation of F at a position which is followed by N_1 and N_2 in the query sequence. For each $i = 1$ *to* 3, $P(F_i|\mathrm{N}_1, \mathrm{N}_2)$ is the probability that $\mathrm{N}_1\mathrm{pN}_2$ was converted from GpN, HpG, and HpH respectively by bisulfite-treatment and sequence error. Using $P(y|d)$, this can be simply calculated by:

$$\begin{aligned} P(F_1|\mathrm{N}_1, \mathrm{N}_2) &= \mathcal{M}_{1\mathrm{N}_1\mathrm{G}} \\ P(F_2|\mathrm{N}_1, \mathrm{N}_2) &= \mathcal{M}_{1\mathrm{N}_1\mathrm{H}} \cdot \mathcal{M}_{2\mathrm{N}_2\mathrm{G}} \\ P(F_3|\mathrm{N}_1, \mathrm{N}_2) &= \mathcal{M}_{1\mathrm{N}_1\mathrm{H}} \cdot \mathcal{M}_{2\mathrm{N}_2\mathrm{H}} \end{aligned}$$

Here:

$$\begin{aligned} \mathcal{M}_i &= \begin{array}{c} \\ H \\ G \end{array} \begin{array}{c} \begin{array}{cc} H & G \end{array} \\ \begin{pmatrix} 1 - u_i B_g & u_i B_g \\ u_i \overline{B_g} & 1 - u_i \overline{B_g} \end{pmatrix} \end{array} \\ &= \begin{array}{c} \\ H \\ G \end{array} \begin{array}{c} \begin{array}{cc} H & G \end{array} \\ \begin{pmatrix} 1 - u_i B_g & u_i B_g \\ e & 1 - e \end{pmatrix} \end{array} \\ \overline{B_g} &= 1 - B_g \\ u_i &= \text{the corresponding } u \text{ for } \mathrm{N_i} \end{aligned}$$

On the other hand, if we focus on reference sequence context, we can calculate a score matrix by using the following, where we distinguish Cs as C_1pGpN, C_2pHpG, and C_3pHpH:

$$\begin{aligned} B'_c &= (1 - F)B_c \\ B'_t &= B_t + FB_c \\ M'_{c_i c} &= (1 - F_i)M_{c_i c} \\ M'_{c_i t} &= M_{c_i t} + F_i M_{c_i c} \\ F &= \frac{A_{c_1}F_1 + A_{c_2}F_2 + A_{c_3}F_3}{A_{c_1} + A_{c_2} + A_{c_3}} \end{aligned}$$

Actually, to calculate these strictly is difficult. The former method has a problem of interdependency between F_{est} and B_y, and the latter method includes nontrivial parameters $M_{c_i c}$. In both cases we need some approximation such as assigning each $M_{c_i c}$ the same value as M_{cc}.

We have not yet implemented these ideas. The query-centric approach could be implemented by converting each query to a position-specific score matrix (PSSM). The reference-centric approach could be implemented by converting the DNA to a six-letter alphabet, with three types of C. (Last accepts PSSM queries and arbitrary alphabets.)

1.5 Aligning bisulfite data with g→a conversions

As mentioned in the main text, there are two variants of bisulfite sequencing. The first produces sequences with c→t conversions. The second produces a mixture of sequences with c→t conversions, and their reverse-complements,

which therefore have g→a conversions. Here, we discuss the second type of data.

If the experiment is performed in a suitable fashion, the DNA reads have "tags" that distinguish bisulfite-converted strands from reverse-complements [1]. In that case, we can align the former using the Last recipe given above, and the latter using this recipe:

```
lastal -p bisulfite_f.mat -s0 -Q1 -j1 -d120 -f0 fIndex reads.fastq > xOut
lastal -p bisulfite_r.mat -s1 -Q1 -j1 -d120 -f0 rIndex reads.fastq > yOut
last-merge-batches.py xOut yOut | last-map-probs.py -s150 -m0.1
```

In this new recipe, the first `lastal` command aligns the reverse-complements of the DNA reads to the genome, allowing for c→t conversions. The second `lastal` command aligns the reads to the genome, allowing for g→a conversions.

The main text describes a risk of biased methylation estimates, which can be avoided by computationally converting all cs in the reads to ts prior to alignment. For reverse-complement reads, we would instead convert gs to as.

For DNA reads without tags, the situation is problematic. We could align each read using all four `lastal` commands, but there is an ambiguity. Suppose that one DNA read is a bisulfite-converted sequence from the reference strand of the genome, and it happens to contain zero unmethylated cytosines. We cannot tell whether it is a bisulfite-converted reference-strand read, or the reverse-complement of a bisulfite-converted non-reference-strand read. Therefore, the read provides evidence for the methylation status of either the reference or the non-reference strand, but we cannot tell which. To make matters worse, if we simply discard such ambiguous reads, we are likely to underestimate methylation rates (because the ambiguity correlates with cytosine methylation). This problem is not specific to Last.

2 Supplementary results

2.1 Effect of the mismatch score

The score matrix that we used with Last (Table 1) is optimal for sequences with ∼99% identity (before bisulfite conversion). However, the human polymorphism rate is closer to 99.9% identity. Therefore, we also tried this matrix:

	a	c	g	t
a	6	-30	-30	-30
c	-30	6	-30	3
g	-30	-30	6	-30
t	-30	-30	-30	3

(When using this matrix, we also set the `lastal` parameter y=150. This aims to prevent the X-drop algorithm from quitting too soon in the face of the stringent mismatch cost.)

Using this score matrix, the accuracy was almost unchanged, and actually marginally worse for dataset B (Figure S1).

It is unclear which of these two matrices is better in practice. The 99%-identity matrix will be more tolerant of over-optimistic base qualities and high

polymorphism rates (while also working quite well for low polymorphism rates). On the other hand, the 99.9%-identity matrix will be less tolerant of wrong alignments involving paralogs or repetitive sequence. Real data likely includes more reads from outside the reference genome (unsequenced regions, alternative haplotypes, contaminants, etc.) than our benchmark: this suggests that a more stringent matrix might be appropriate.

2.2 Effect of integer-rounding with quality data

The final step in Last's procedure for using sequence quality data, after calculating S'_{xd}, is to round S'_{xd} to the nearest integer [2]. It is conceivable that this rounding harms Last's accuracy.

To examine this issue, we re-ran Last after multiplying all its score parameters by 10, which reduces the impact of rounding. In addition to scaling the score matrix, we set these `lastal` parameters: y=440, d=1200; and this `last-map-probs` parameter: s=1500. These scaled scores caused almost no change in accuracy (Figure S1).

2.3 Effect of bases with phred score 2

Our datasets have many bases with phred score 2 (Figure 1). This is because Illumina data uses 2 to indicate nonspecific errors rather than error probability $10^{-2/10} = 0.63$. However, in our benchmark these bases actually have error probability 0.63, and moreover Last assumes that they have this error probability. It might be argued that this gives Last an unrealistic advantage.

To address this concern, we re-ran Last after trimming the 3' end of each read to just before the first base with phred score < 3. For dataset A, this decreased the accuracy by only a tiny amount (Figure S1). For dataset B, surprisingly, it increased the accuracy.

We do not understand the latter result, but we suspect it relates to the fact that, for most wrongly-aligned reads, the correct alignment was not found by `lastal` (as opposed to cases where it was found by `lastal` but wrongly evaluated by `last-map-probs`: see Supplementary dataset 1). In other words, most errors are due to the alignment-search heuristic and not the scoring scheme.

2.4 Effect of the alignment score threshold

With our standard settings, `lastal` reports alignments with score ≥ 120, and then `last-map-probs` reports alignments with score ≥ 150.

Score	E-value
180	0.222
150	232
120	239000

The reason for using a threshold of 150 is that this is high enough to make chance alignments rare. Specifically, if we aligned 1 million random length-85 sequences, with the same base frequencies as our bisulfite-converted reads, against a randomized genome, we would expect $\sim$232 alignments with score $\geq$ 150. (We calculated this using `lastex`.)

The reason why `lastal` uses a score threshold of 120 is to enable `last-map-probs` to estimate alignment probabilities accurately. If one DNA read aligns to two locations, and the alignment scores differ by 30, the probabilities will differ by a factor of 1000. (This is because the scores are phred-scaled, so 30 represents 3 powers of 10.) Thus, by considering scores as low as 120, `last-map-probs`'s probabilities should be accurate to about 1 part in 1000.

We also tried using a `last-map-probs` score threshold of 180, without changing the threshold for `lastal`. This has two effects: it makes chance alignments rarer, and it makes `last-map-probs`'s probabilities accurate to 1 part in a million. In our benchmark, this produced inferior results: the error rate decreased slightly, but this was outweighed by a decrease in sensitivity (Figure S2).

Real data likely includes more contaminants and artifacts than our benchmark, and it is possible that a higher score threshold would then be beneficial.

2.5 Effect of the seed pattern

Aside from contiguous seeds (Figure 4), we tried four seed patterns (Figure S3). These patterns were taken from Last's documentation (`tag-seeds.txt`): the first two are tuned for 1-mismatch hits, and the second two are tuned for 2-mismatch hits. This is by no means a comprehensive analysis. In our benchmark, the 2-mismatch patterns gave slightly better accuracy (Figure S3).

Note that classic seed patterns (such as from [5]) are unlikely to be suitable, because they are tuned for remote homology search whereas we are dealing with high-similarity alignments.

References

[1] P. Y. Chen, S. J. Cokus, and M. Pellegrini. BS Seeker: precise mapping for bisulfite sequencing. *BMC Bioinformatics*, 11:203, 2010.

[2] M. C. Frith, R. Wan, and P. Horton. Incorporating sequence quality data into alignment improves DNA read mapping. *Nucleic Acids Res.*, 38:e100, Apr 2010.

[3] M. Hamada, E. Wijaya, M. C. Frith, and K. Asai. Probabilistic alignments with quality scores: an application to short-read mapping toward accurate SNP/indel detection. *Bioinformatics*, 27:3085–3092, Nov 2011.

[4] R. Lister, M. Pelizzola, R. H. Dowen, R. D. Hawkins, G. Hon, J. Tonti-Filippini, J. R. Nery, L. Lee, Z. Ye, Q. M. Ngo, L. Edsall, J. Antosiewicz-Bourget, R. Stewart, V. Ruotti, A. H. Millar, J. A. Thomson, B. Ren, and J. R. Ecker. Human DNA methylomes at base resolution show widespread epigenomic differences. *Nature*, 462:315–322, Nov 2009.

[5] B. Ma, J. Tromp, and M. Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18:440–445, Mar 2002.

Supplementary figures

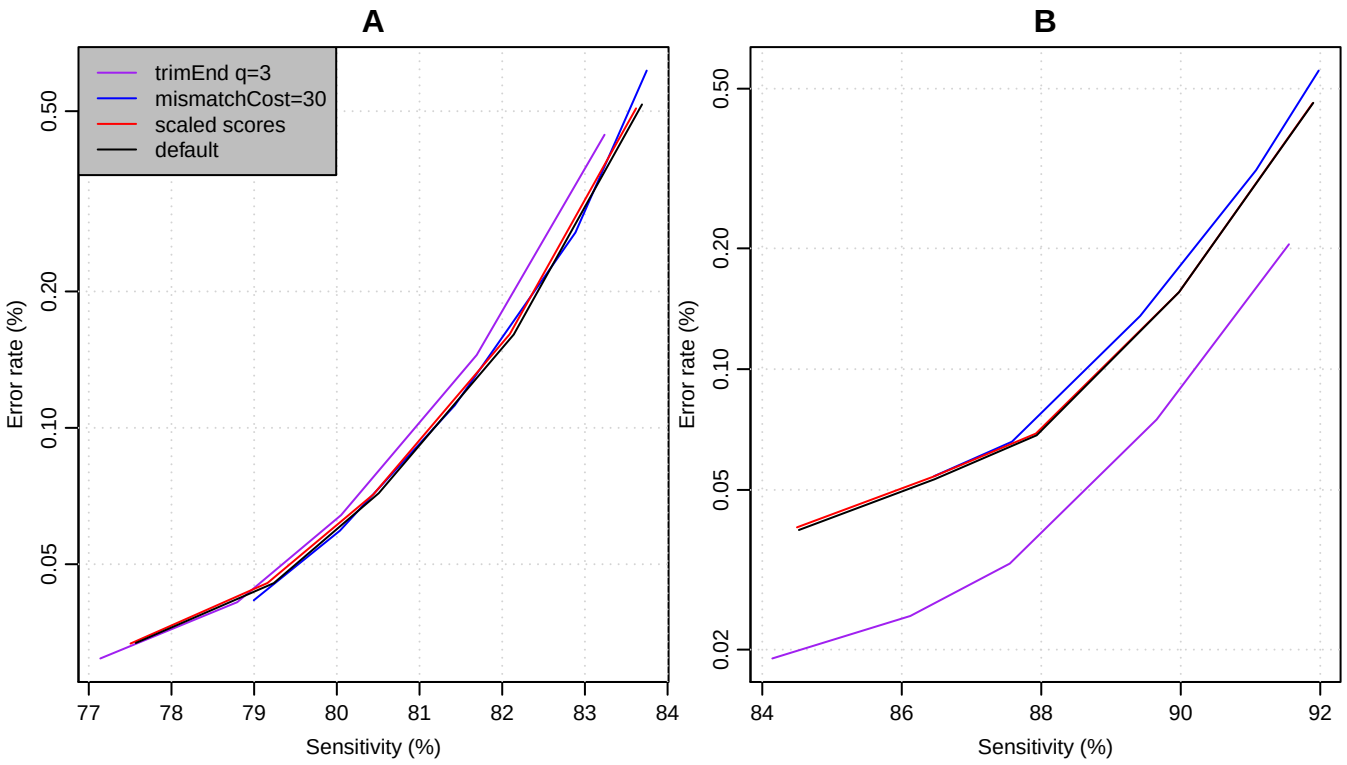

Figure S1: Accuracy of LAST, with various parameter settings, for aligning bisulfite-converted DNA reads to the reference genome. The two panels refer to datasets (A) and (B). The black lines in this figure are identical to those in Figure 2.

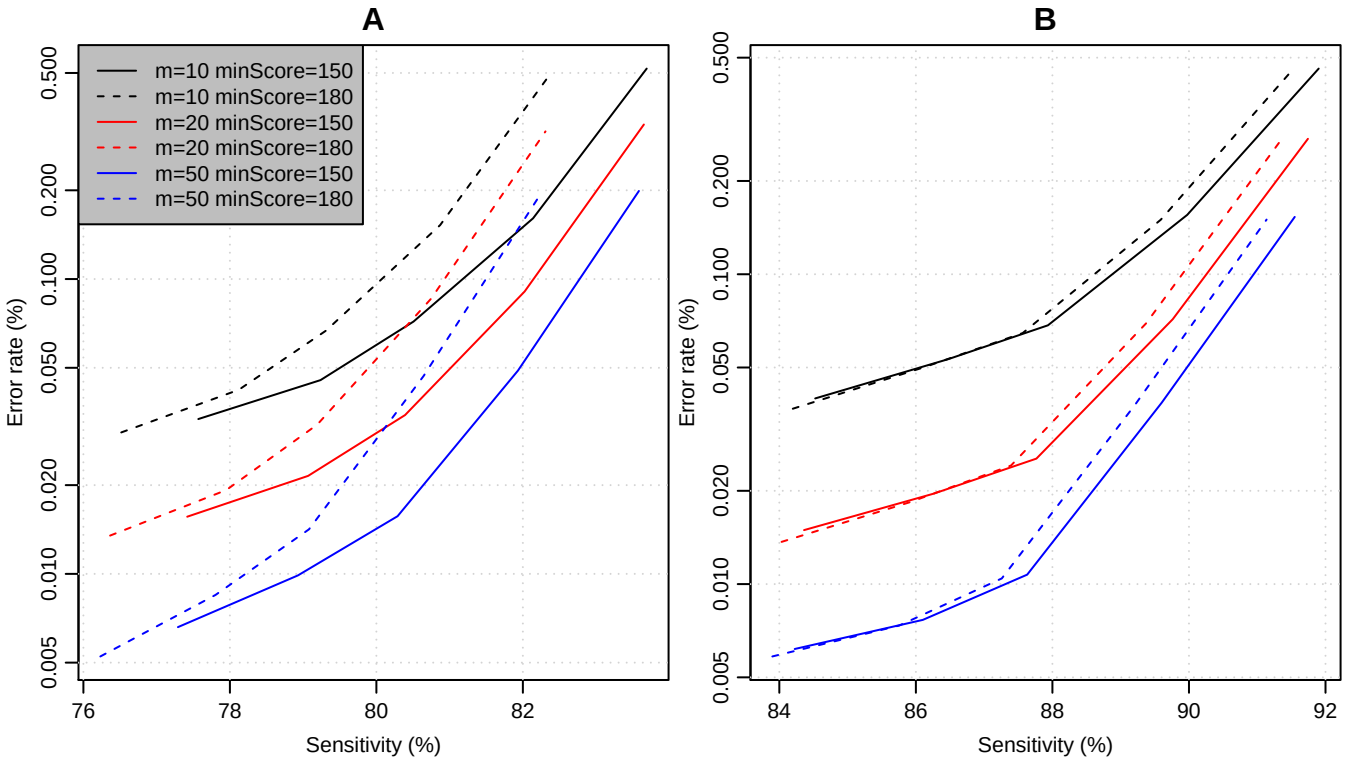

Figure S2: Accuracy of LAST, with different m and minScore settings, for aligning bisulfite-converted DNA reads to the reference genome. The two panels refer to datasets (A) and (B). The black lines in this figure are identical to those in Figure 2.

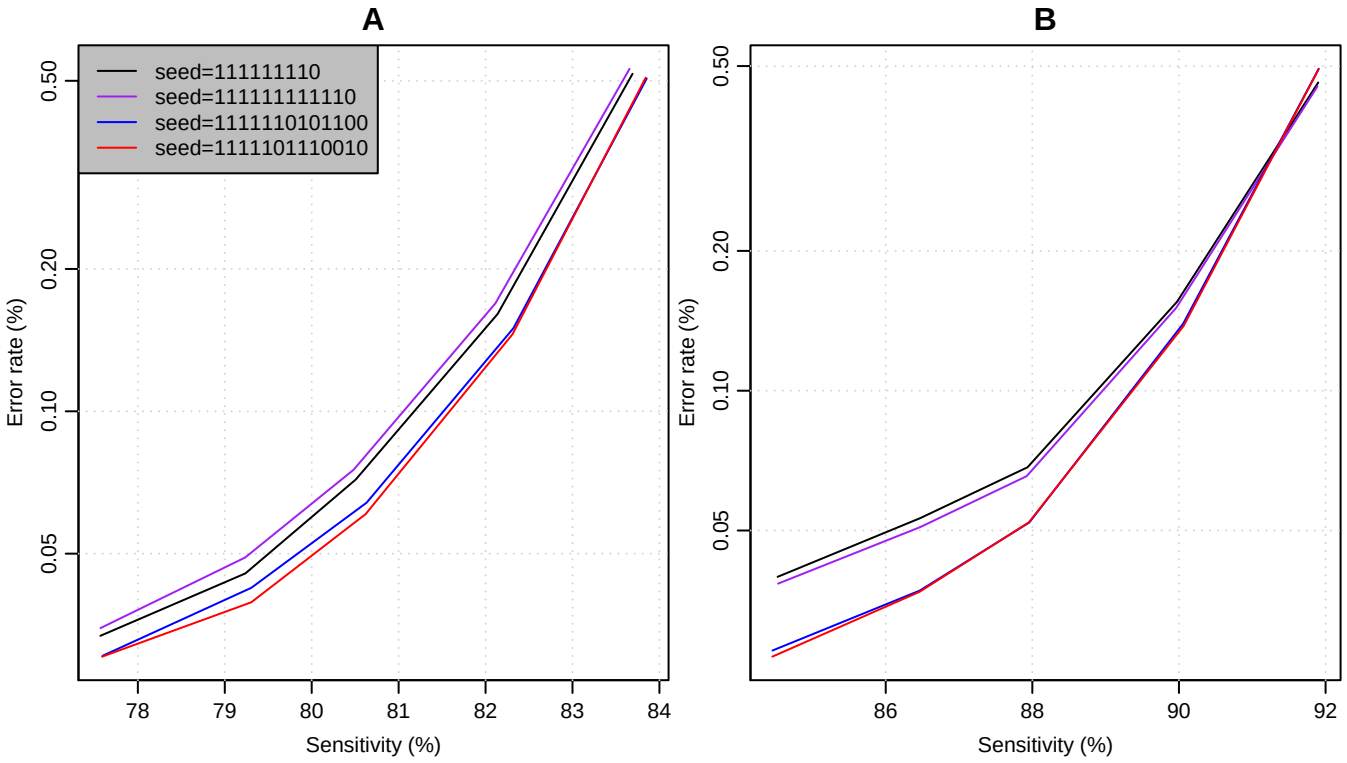

Figure S3: Accuracy of LAST, with different seed patterns, for aligning bisulfite-converted DNA reads to the reference genome. The two panels refer to datasets (A) and (B). The black lines in this figure are identical to those in Figure 2.

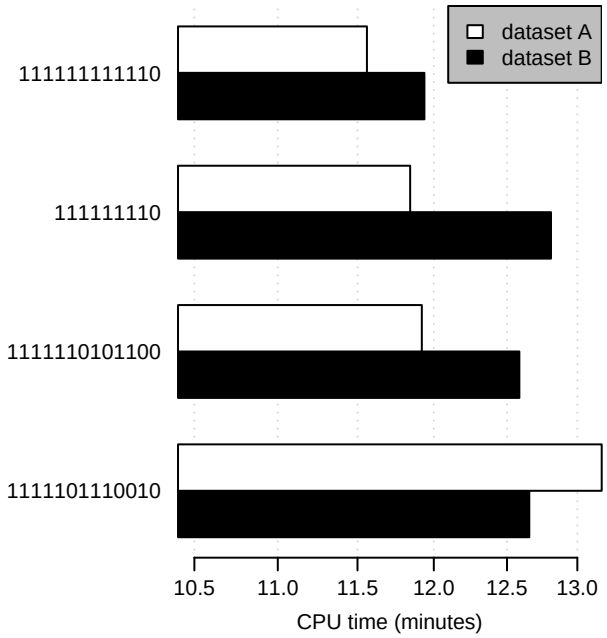

Figure S4: Run times for aligning bisulfite-converted DNA reads to the reference genome with LAST, using different seed patterns.